Opérateur NORM_MODE

1 But

Le rôle de la commande est de normer des modes propres en fonction d'un critère choisi par l'utilisateur.

Les opérateurs de calcul modal MODE_ITER_INV [U4.52.04] et MODE_ITER_SIMULT [U4.52.03] produisent un concept de type mode_meca ou mode_meca_c dont les modes propres réels ou complexes sont normalisés de telle façon que la plus grande des composantes qui n'est pas un multiplicateur de LAGRANGE, soit égale à 1.

L'opérateur NORM_MODE permet à l'utilisateur de choisir une autre méthode de normalisation, par exemple masse généralisée, rigidité généralisée ...

En fonction de la normalisation choisie, les paramètres modaux (facteur de participation, masse effective, ...) sont réactualisés.

Opérateur ré-entrant.

2 Syntaxe

```
m_out = NORM_MODE (

        ◊  reuse      =  m_out ,
        ♦  MODE       =  m_in ,                      /  [mode_meca]
                                                     /  [mode_meca_c]
                                                     /  [mode_flamb]

        ◊  /  NORME      =  /   'MASS_GENE',
                            /   'RIGI_GENE',
                            /   'TRAN',
                            /   'TRAN_ROTA',
                            /   'EUCL',
                            /   'EUCL_TRAN',
           /  ◊  NOEUD     =  no,                       [noeud]
              ◊  GROUP_NO  =  grno,                     [group_no]
               # Si NOEUD ou GROUP_NO
              ♦  NOM_CMP   =  cmp,                      [Kn]
           /  SANS_CMP  =  s_cmp,                       [l_Kn]
           /  AVEC_CMP  =  a_cmp,                       [l_Kn]

        ◊  MODE_SIGNE   =  _F  (
              ♦  / NOEUD    =  no,                      [noeud]
                 / GROUP_NO=  grno,                     [group_no]
              ♦  NOM_CMP   =  cmp,                      [Kn]
              ◊  SIGNE  =  /   'POSITIF',               [DEFAUT]
                           /   'NEGATIF',
                           )

        ◊  MASSE      =  masse ,                        [matr_asse_depl_r]
                                                     ou [matr_asse_gene_r]
                                                     ou [matr_asse_pres_r]
        ◊  RAIDE      =  masse ,                        [matr_asse_depl_r]
                                                     ou [matr_asse_depl_c]
                                                     ou [matr_asse_gene_r]
                                                     ou [matr_asse_pres_r]
        ◊  AMOR       =  masse ,                        [matr_asse_depl_r]
                                                     ou [matr_asse_gene_r]

        ◊  TITRE  =      t,                             [l_Kn]

        ◊  INFO   =      /  1,                          [DEFAUT]
                         /  2,

                                );

        si m_in est de type [ mode_meca ]
        alors m_out est de type [ mode_meca ]
        idem avec [ mode_meca_c ]
        idem avec [ mode_flamb ]
```

3 Opérandes

3.1 Opérande MODE

♦ MODE = m_in

Nom du concept de type mode_* dont on veut changer la normalisation des modes propres. Si m_out est identique à m_in et si le mot-clé 'reuse' est activé avec la valeur m_out, la renormalisation se fait en place.

3.2 Opérande NORME

◊ / NORME =

Nom symbolique de la norme choisie.

'MASS_GENE' :

Les modes sont normalisés à la masse généralisée unitaire.

'RIGI_GENE' :

Les modes sont normés à la rigidité généralisée unitaire.

'TRAN' :

Les modes sont normés à 1. pour la plus grande des composantes de translation : (composantes : DX, DY, DZ).

'TRAN_ROTA' :

Les modes sont normés à 1. pour la plus grande des composantes de translation et de rotation (composantes : DX, DY, DZ, DRX, DRY, DRZ).

'EUCL' :

Les modes sont normalisés à la norme euclidienne des composantes qui ne sont pas des multiplicateurs de LAGRANGE (composante : LAGR).

'EUCL_TRAN' :

Les modes sont normalisés à la norme euclidienne des composantes qui sont des composantes de translation (composantes : DX, DY, DZ).

3.3 Opérandes NOEUD ou GROUP_NO et NOM_CMP

◊ / NOEUD = no
/ GROUP_NO = grno

Nom du noeud no ou du groupe d'un nœud $grno$ où on normalise.

| *Attention : le groupe de nœuds $grno$ doit contenir un unique nœud.*

♦ NOM_CMP = cmp

Nom de la composante de normalisation au nœud no ou au groupe d'un nœud $grno$. Cet opérande est obligatoire si NOEUD ou GROUP_NO est renseigné.

Les modes sont normés à $1.$ pour la composante cmp du nœud no ou du groupe d'un nœud $grno$.

3.4 Opérandes AVEC_CMP / SANS_CMP

◊ / AVEC_CMP = a_cmp

a_cmp liste des noms des composantes utilisées pour la normalisation.

Les modes sont normés à 1. pour la plus grande des composantes de la liste a_cmp quelque soit le nœud.

/ SANS_CMP = s_cmp

s_cmp liste des noms des composantes qui ne sont pas utilisées pour la normalisation. Les modes sont normés à 1. pour la plus grande des composantes qui n'est pas dans la liste s_cmp.

3.5 Mot-clé facteur MODE_SIGNE

Ce mot-clé facteur permet d'imposer pour tous les modes le signe d'une composante d'un noeud stipulée par l'utilisateur. Ce mot-clé facteur ne peut être utilisé que pour les modes réels (problème généralisé).

♦ / NOEUD = nd
/ GROUP_NO = grnd

Nom du nœud ou du groupe d'un noeud où on impose le signe d'une composante.

| *Attention : le groupe de nœuds $grnd$ doit contenir un unique nœud.*

♦ NOM_CMP = cmp

Nom de la composante du nœud nd ou du groupe d'un nœud $grnd$ où le signe est imposé.

◊ SIGNE = / 'POSITIF'
/ 'NEGATIF'

Signe imposé de la composante : 'POSITIF' ou 'NEGATIF'.

3.6 Cas de la normalisation d'une collection de modes issue de DEFI_BASE_MODALE

Si on veut normer une collection de modes (base modale) issue de DEFI_BASE_MODALE , il faut renseigner les deux matrices, de masse et de raideur, permettant d'actualiser les paramètres modaux :

```
◊ MASSE = masse          [matr_asse_depl_r]
                      ou [matr_asse_gene_r]
                      ou [matr_asse_pres_r]
◊ RAIDE = masse          [matr_asse_depl_r]
                      ou [matr_asse_depl_c]
                      ou [matr_asse_gene_r]
                      ou [matr_asse_pres_r]
```

En effet, dans ce cas, les informations sur les matrices de masse et de rigidité (et éventuellement d'amortissement) sur lesquelles s'appuie la base modale ont été perdues, ou les modes peuvent être issus de différents jeux de matrices. Il est donc nécessaire de les rappeler à l'opérateur NORM_MODE.

Dans le cas d'une base de modes complexes, il faut de plus donner une matrice d'amortissement si on veut normer par rapport à la masse généralisée ou à la raideur généralisée (cf paragraphe 4.2).

```
◊ AMOR = masse           [matr_asse_depl_r]
                      ou [matr_asse_gene_r]
```

3.7 Opérande TITRE

◊ TITRE = t

Titre associé au concept produit par cet opérateur [U4.03.01].

3.8 Opérande INFO

◊ INFO = 1 ou 2

Pour chaque mode, le nom de l'ancienne norme et le nom de la nouvelle norme sont indiqués dans le fichier MESSAGE . Les noms imprimés des normes correspondent aux mot-clés décrits aux paragraphes 3.2 , 3.3 , 3.4 .

4 Formulation des règles de normalisation

Les différentes normes utilisées ainsi que la définition des différents paramètres modaux sont recensées dans la documentation de référence [R5.01.03].

4.1 Modes propres réels

Pour les modes de type `mode_meca_r` (modes propres réels) le problème généralisé aux valeurs propres associé est : $(K-\omega^2 M)x=(K-(2\pi f)^2 M)x=0$
où K, M sont respectivement la matrice de masse et la matrice de rigidité du système mécanique.

Pour les modélisations '`MECANIQUE`', on définit les composantes du vecteur propre :

- composantes de translation u^T
- composantes de rotation u^R
- composantes des multiplicateurs de `LAGRANGE` λ
- autres composantes (pression et potentiel fluide) p_f

On appelle :

- u^{TR} composantes de translation et rotation,
- u composantes autres que multiplicateurs de LAGRANGE.

ce qui conduit à :

$$u^*=\begin{bmatrix} u \\ \lambda \end{bmatrix}=\begin{bmatrix} u^T \\ u^R \\ p_f \\ \lambda \end{bmatrix}$$

Pour les modèles avec composantes de translation et de rotation, le mode propre ϕ_i fourni par les algorithmes d'analyse modale est par défaut :

$$\Phi_i=\frac{u^*}{max\, u}=\frac{u^*}{max\, u^{TR}}=\Phi_i^{TR}$$

ce qui est équivalent à la normalisation obtenue par le mot clé '`TRAN_ROTA`'.

Avec le mot clé '`TRAN`' le mode obtenu est défini par :

$$\Phi_i=\frac{u^*}{max\, u^T}=\Phi_i^T$$

Pour les modèles avec composantes de translation uniquement, la normalisation est par défaut :

$$\Phi_i^T=\frac{u^*}{max\, u}=\frac{u^*}{max\, u^T}$$

ce qui est équivalent à la normalisation obtenue par le mot clé '`TRAN`'.

La normalisation par défaut conduit aux paramètres généralisés suivants :

- rigidité généralisée ${}^T\Phi_i K \Phi_i = \gamma_i$
- masse généralisée ${}^T\Phi_i M \Phi_i = \mu_i$
- d'où la pulsation propre $\omega_i^2 = \frac{\gamma_i}{\mu_i}$

La normalisation à la masse généralisée unitaire est obtenue par le mot clé 'MASS_GENE' :

$$\Phi_i^M = \frac{\Phi_i}{\sqrt{\mu_i}} \text{ d'où } {}^T\Phi_i^M M \Phi_i^M = 1. \text{ et } {}^T\Phi_i^M K \Phi_i^M = \omega_i^2$$

Celle à la rigidité généralisée unitaire est obtenue par le mot clé 'RIGI_GENE' :

$$\Phi_i^K = \frac{\Phi_i}{\sqrt{\gamma_i}} \text{ d'où } {}^T\Phi_i^K M \Phi_i^K = \frac{1}{\omega_i^2} \text{ et } {}^T\Phi_i^K K \Phi_i^K = 1.$$

La normalisation du mode propre à la norme euclidienne 'EUCL' est obtenue naturellement par :

$$\Phi_i^{\|u\|} = \frac{u^*}{\|u\|} = \frac{u^*}{\sqrt{\sum_j (u_j)^2}}$$

La normalisation du mode propre à la norme euclidienne 'EUCL_TRAN' est :

$$\Phi_i^{\|u^T\|} = \frac{u^*}{\|u^T\|} = \frac{u^*}{\sqrt{\sum_j (u_j^T)^2}}$$

4.2 Modes propres complexes

Pour les modes de type mode_meca_c (modes propres complexes) issus d'une résolution d'un problème quadratique aux valeurs propres $\lambda^2 M + \lambda C + K = 0$ où C est la matrice d'amortissement du système mécanique, on norme les modes Φ par rapport au problème linéarisé associé :

$$\left(\lambda \begin{bmatrix} 0 & M \\ M & C \end{bmatrix} + \begin{bmatrix} -M & 0 \\ 0 & K \end{bmatrix}\right) \begin{pmatrix} \lambda \Phi \\ \Phi \end{pmatrix} = 0$$

Le mode propre est normé à la masse généralisée unitaire ('MASS_GENE'), si Φ_i satisfait :

$$\left(\lambda\, {}^T\Phi_i \;\; {}^T\Phi_i\right) \begin{bmatrix} 0 & M \\ M & C \end{bmatrix} \begin{pmatrix} \lambda \Phi_i \\ \Phi_i \end{pmatrix} = 1.$$

à la rigidité généralisée unitaire ('RIGI_GENE'), si Φ_i satisfait :

$$\left(\lambda\, {}^T\Phi_i \;\; {}^T\Phi_i\right) \begin{bmatrix} -M & 0 \\ 0 & K \end{bmatrix} \begin{pmatrix} \lambda \Phi_i \\ \Phi_i \end{pmatrix} = 1.$$

Pour les autres normes, les définitions sont équivalentes à celles définies pour les modes réels, il suffit de remplacer le produit scalaire par le produit hermitien.

5 Paramètres modaux mis à jour

Un concept de type `mode_meca` peut contenir, pour chaque mode, les paramètres modaux suivants (visibles par exemple en imprimant le concept avec la commande `IMPR_RESU` au `FORMAT='RESULTAT'` avec l'option `TOUT_PARA='OUI'`) :

Intitulé du paramètre dans Code_Aster	Définition
FREQ	Fréquence propre (amortie, le cas échéant)
AMOR_GENE	Amortissement modal généralisé
AMOR_REDUIT	Amortissement modal réduit
FACT_PARTICI_D* (* = X ou Y ou Z)	Facteur de participation du mode dans la direction D*
MASS_EFFE_D* (* = X ou Y ou Z)	Masse modale effective dans la direction D*
MASS_EFFE_UN_D* (* = X ou Y ou Z)	Masse modale effective unitaire dans la direction D*
MASS_GENE	Masse généralisée du mode
OMEGA2	Pulsation propre (amortie, le cas échéant) au carré
RIGI_GENE	Raideur généralisée du mode

Tableau 5.1 : liste des paramètres modaux.

Ces paramètres sont définis mathématiquement dans la documentation de référence [R5.01.03].

L'opérateur `NORM_MODE` calcule ou met à jour les paramètres modaux suivants, qui dépendent de la normalisation choisie : `FACT_PARTICI_D*`, `MASS_GENE` et `RIGI_GENE`. Il enrichit également la structure de donnée avec les paramètres `MASS_EFFE_UN_D*` (indépendants de la normalisation).
Les autres paramètres sont indépendants de la normalisation.

6 Exemples pour des modes réels

Pour les modes de type `mode_meca` (modes propres réels) issus d'une résolution d'un problème généralisé aux valeurs propres $(K-\lambda M)x=0$:

- normer un vecteur propre x à la rigidité généralisée unitaire équivaut à ce que x satisfasse

  $$x^T K x=1$$

  Normalisation avec duplication du concept `mode_meca` :

```
mo_2    = NORM_MODE  (   MODE = mo_1,
                         NORME= 'RIGI_GENE'
                     );
```

- normer un vecteur propre x à la masse généralisée unitaire équivaut à ce que x satisfasse

  $$x^T M x=1$$

  Normalisation à la masse généralisée unitaire, avec écrasement du concept `mode_meca` :

```
mo   = NORM_MODE         (   reuse = mo,
                             MODE  = mo,
                             NORME = 'MASS_GENE'
                         );
```